Design Overview for *"Live Map Of International Space Station"*

Name: Thanh Tam Vo
Student ID: 103487596

Summary of Program

This program is an updated version of the project "International Space Station Tracking", the program will show the current position of the International Space Station of NASA (This program can be considered as a live map of ISS). Also, it will show the trajectory of the ISS from the beginning.

The position of ISS will be presented in the following map below

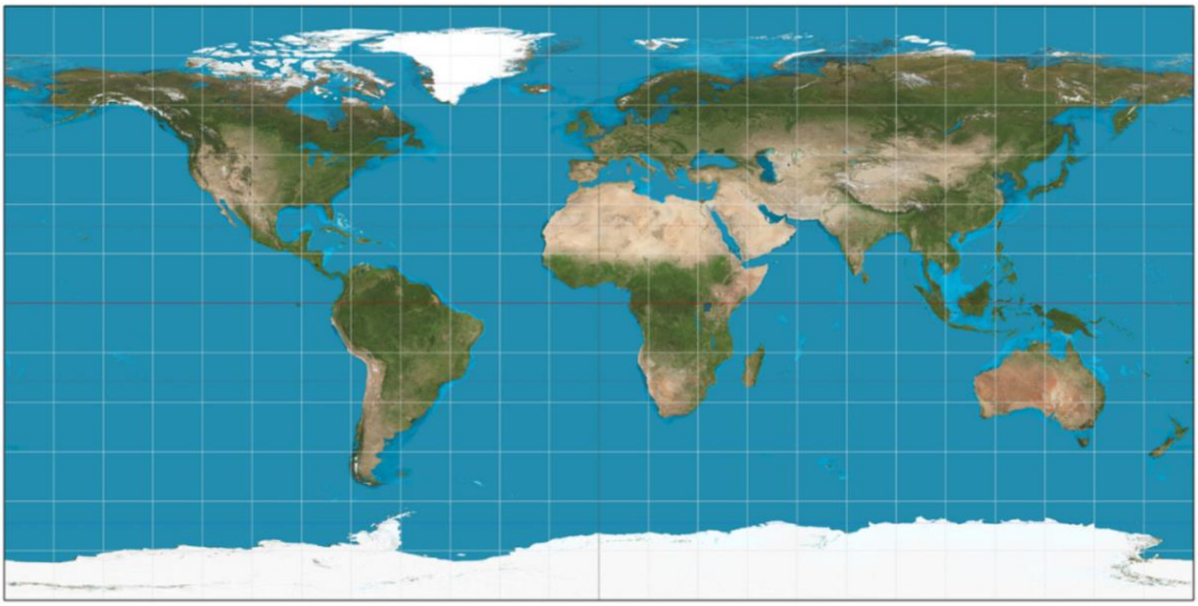


Figure 1. Earth map of the program

Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: Object_Management.cs

Responsibility	**Type Details**	**Notes**
Managing application objects by their IDs	List<string> : _listOfObject	

Table 2: Application_Object.cs

Responsibility	Type Details	Notes
Providing common features of the application objects	name : string isHuman: bool	This is an abstract class It also inherits from the Object_Management class

Table 3: Craft.cs

Responsibility	Type Details	Notes
Demonstrating the International Space Station object in the program.	longitude : string latitude : string	It inherits from the Application_Object class

Table 4: Astronaut.cs

Responsibility	Type Details	Notes
Demonstrating the astronaut object in the program.	craft : Craft	It inherits from the Application_Object class

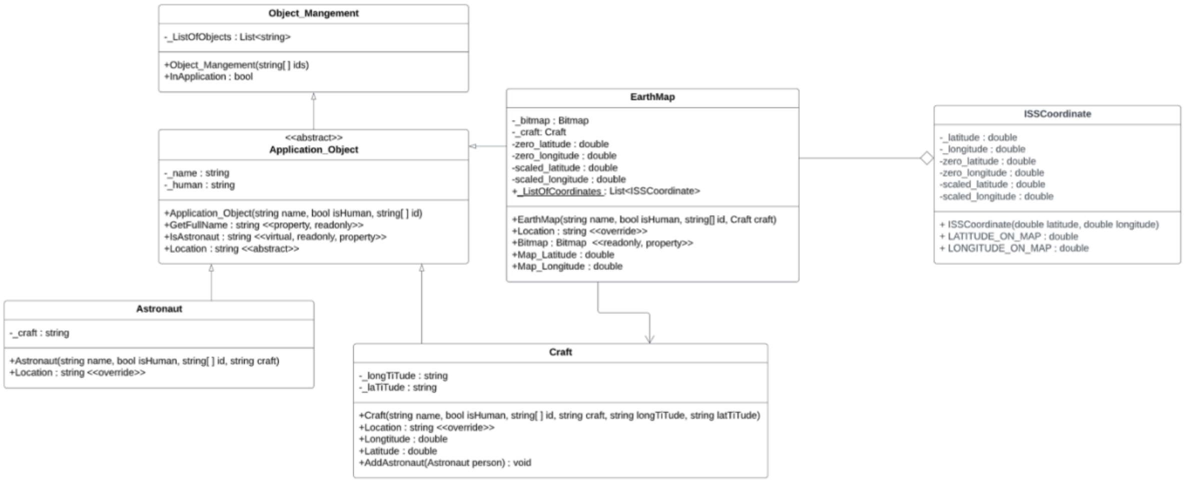
Table 5: EarthMap.cs

Responsibility	Type Details	Notes
Demonstrating the earth map object in the program	bitmap : Bitmap craft : Craft _ListOfCoordinates : List<ISSCoordinate>	It inherits from the Application_Object class

Table 6: ISSCoordinate.cs

Responsibility	Type Details	Notes
Demonstrating the coordinate system (involving latitude and longitude)	_latitude : double _longitude : double	Do not inherit from any class

Expected UML Diagram


Object_Mangement
-_ListOfObjects : List<string>
+Object_Mangement(string[ ] ids)
+InApplication : bool
<<abstract>>
Application_Object
-_name : string
-_human : string
+Application_Object(string name, bool isHuman, string[ ] id)
+GetFullName : string <<property, readonly>>
+IsAstronaut : string <<virtual, readonly, property>>
+Location : string <<abstract>>
Astronaut
-_craft : string
+Astronaut(string name, bool isHuman, string[ ] id, string craft)
+Location : string <<override>>
EarthMap
-_bitmap : Bitmap
-_craft: Craft
-zero_latitude : double
-zero_longitude : double
-scaled_latitude : double
-scaled_longitude : double
+_ListOfCoordinates : List<ISSCoordinate>
+EarthMap(string name, bool isHuman, string[] id, Craft craft)
+Location : string <<override>>
+Bitmap : Bitmap <<readonly, property>>
+Map_Latitude : double
+Map_Longitude : double
ISSCoordinate
-_latitude : double
-_longitude : double
-zero_latitude : double
-zero_longitude : double
-scaled_latitude : double
-scaled_longitude : double
+ ISSCoordinate(double latitude, double longitude)
+ LATITUDE_ON_MAP : double
+ LONGITUDE_ON_MAP : double
Craft
-_longTiTude : string
-_laTiTude : string
+Craft(string name, bool isHuman, string[ ] id, string craft, string longTiTude, string latTiTude)
+Location : string <<override>>
+Longtitude : double
+Latitude : double
+AddAstronaut(Astronaut person) : void

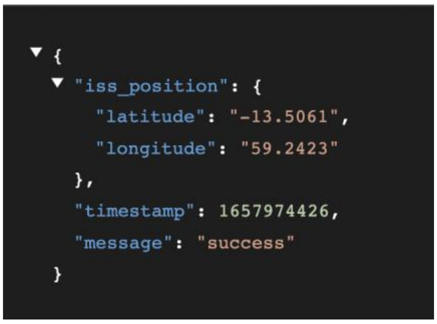
(Screenshot from LucidChart on July 23th, 2022)

Overview of program structure

The program show the current position of the ISS and its trajectory for analysis purposes

***Note:** This program uses Pure Fabrication concept by storing the JSON data in a temporary class.

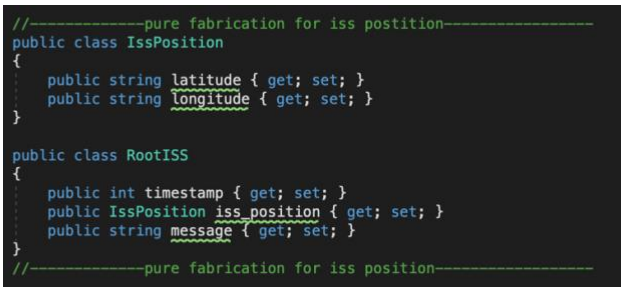
The JSON format of the International Space Station does not change, according to Open Notify (open-notify.org), the JSON format of ISS is shown below



```
▼ {
  ▼ "iss_position": {
      "latitude": "-13.5061",
      "longitude": "59.2423"
    },
    "timestamp": 1657974426,
    "message": "success"
  }
```

JSON format of ISS
(Captured from Open Notify)

Therefore, inside the Main program, I created two temporary class to store the data above



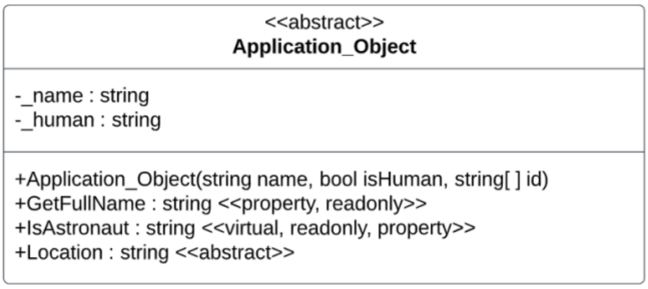
```
//-------------pure fabrication for iss postition-----------------
public class IssPosition
{
    public string latitude { get; set; }
    public string longitude { get; set; }
}

public class RootISS
{
    public int timestamp { get; set; }
    public IssPosition iss_position { get; set; }
    public string message { get; set; }
}
//-------------pure fabrication for iss position------------------
```

The application of pure fabrication in the program
(Captured from my Visual Studio)

Abstraction in program

The program will have an abstract class, which is the class called *Application_Object,* by indicating the common feature of the application object such as: name, is it a human or not, its location, etc.



<<abstract>>
Application_Object

-_name : string
-_human : string

+Application_Object(string name, bool isHuman, string[ ] id)
+GetFullName : string <<property, readonly>>
+IsAstronaut : string <<virtual, readonly, property>>
+Location : string <<abstract>>

Application_Object abstract class
(Captured from LucidChart)

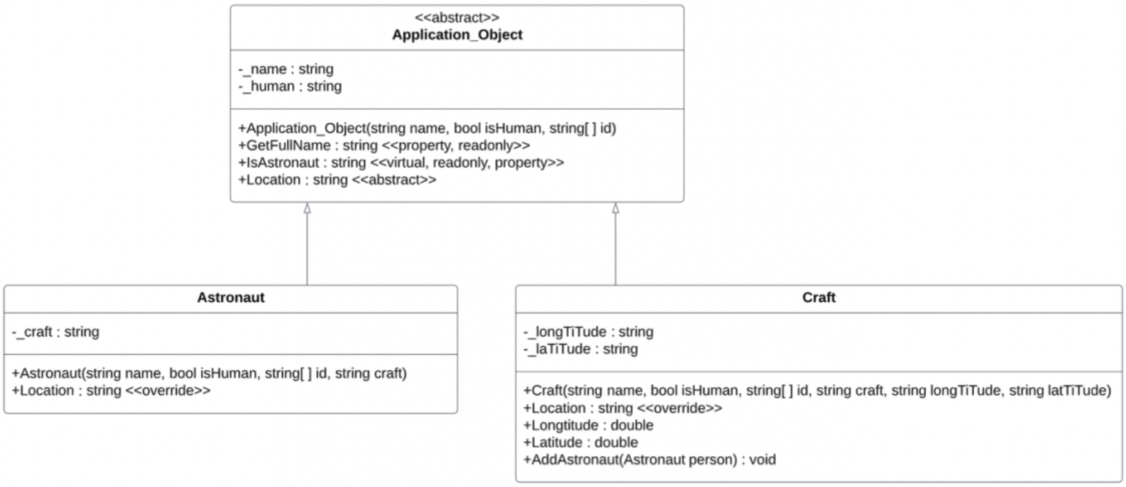
Inheritance and Polymorphism in program

There are two classes that inherit from the *Application_Object* class, they are *Craft* and *Astronaut* because both are main objects of this console application.

Both have the *Location()* function (because they inherit from the *Application_Object* class), however, each class performs differently.

+The *Location()* function of the *Craft* class will return the craft's latitude and longitude
+ The *Location()* function of the *Astronaut* class will return the astronaut's current craft

The UML diagram for those classes is depicted below


<<abstract>>
Application_Object
-_name : string
-_human : string
+Application_Object(string name, bool isHuman, string[ ] id)
+GetFullName : string <<property, readonly>>
+IsAstronaut : string <<virtual, readonly, property>>
+Location : string <<abstract>>
Astronaut
-_craft : string
+Astronaut(string name, bool isHuman, string[ ] id, string craft)
+Location : string <<override>>
Craft
-_longTiTude : string
-_laTiTude : string
+Craft(string name, bool isHuman, string[ ] id, string craft, string longTiTude, string latTiTude)
+Location : string <<override>>
+Longtitude : double
+Latitude : double
+AddAstronaut(Astronaut person) : void

UML diagram to demonstrate the use of Inheritance and Polymorphism
(Captured from LucidChart)